The Linear Geometry Structure of Label Matrix for Multi-label Learning

Tianzhu Chen[1,2], Fenghua Li[1,2], Fuzhen Zhuang[3,4], Yunchuan Guo[1,2], and Liang Fang[1⋆]

[1] Institute of Information Engineering, Chinese Academy of Sciences, China
[2] School of Cyber Security, University of Chinese Academy of Sciences, China
[3] Key Lab of Intelligent Information Processing of Chinese Academy of Sciences (CAS), Institute of Computing Technology, CAS, Beijing 100190, China
[4] Xiamen Data Intelligence Academy of ICT, CAS, China

Abstract. Multi-label learning annotates a data point with the relevant labels. Under the low-rank assumption, many approaches embed the label space into the low-dimension space to capture the label correlation. However these approaches usually have weak prediction performance because the low-rank assumption is usually violated in real-world applications. In this paper, we observe the fact that the linear representation of row and column vectors of label matrix does not depend on the rank structure and it can capture the linear geometry structure of label matrix (LGSLM). Inspired by the fact, we propose the LGSLM classifier to improve the prediction performance. More specifically, after rearranging the columns of a label matrix in decreasing order according to the number of positive labels, we capture the linear representation of the row vectors of the compact region in the label matrix. Moreover, we also capture the linear and sparse representation of column vectors using the L_1-norm. The experimental results for five real-world datasets show the superior performance of our approach compared with state-of-the-art methods.

Keywords: Multi-label Learning · Linear Representation · Sparse Representation.

1 Introduction

Multi-label learning [12, 18, 20] usually annotates a data object with multiple relevant labels. For example, a web page describing the Divine Comedy may be annotated with the tags "14th-century Christian texts" and "epic poems in Italian"; an article that discusses physical education can be annotated with education and fitness tags; an image that shows a beach and a city can be annotated with beach and city tags. Presently, multi-label learning is applied to various applications ranging from web page retrieval to document classification and image annotation.

⋆ Corresponding author: Liang Fang(fangliang@iie.ac.cn).

Considerable literature has showed that a better prediction performance can be achieved in multi-label learning through capturing label correlation [12, 18, 20]. Based on the rank structure of label matrix, researchers usually exploit the embedding method [2, 5, 11, 18, 21, 22] to capture the label correlation. Generally, the embedding method can be divided into two categories: the linear embedding method and the nonlinear embedding method. The key idea of the linear approach [11, 18, 21, 22] is to exploit a low-rank transform to project the high-dimensional label space onto a low-dimensional space. Subsequently, the prediction is performed on the low-dimensional label space and then the original labels are recovered. However the label correlation learned by the linear embedding method does not capture the intrinsic structure among the labels because the low-rank assumption is often violated in practice.

To address this problem, after grouping all instances into the clusters, Bhatia et al. [2] exploited nonlinear embedding to compress the label matrix of every cluster into a small-dimension label space. By preserving the pairwise distances between a label vector and the nearest label vectors, [2] compressed the label space into a small label space and then learned the regressor in the embedding space. By leveraging the fact that nearest neighbors have been preserved during training, [2] used a k-nearest neighbor (kNN) classifier in the embedding space to predict test instances. Based on the framework of [2], [5] exploited the Skip-Gram Negative Sampling approach to embed the label space of every cluster into low-dimension space. However, a small dimension [2, 5] was used to embed the label space that ignored the overall structure of the label matrix, thereby leading to information loss.

By observing the fact that the linear representation of row and column vectors of label matrix does not depend on the rank structure and it can capture the linear geometry structure of label matrix (LGSLM), in this paper, we propose the LGSLM classifier to capture the more intrinsic geometry structure of the label matrix. More specifically, after rearranging the columns in decreasing order according to the number of 1-valued entries, we segment the label matrix into multiple regions and select the compact region to capture the row representation. In the column representation, we exploit the L_1 norm to constrain the linear representation, which captures the sparse representation. Further, we adjust the label value to capture the linear structure of larger region. The experimental results for five real-world datasets show the superior performance of our approach. The details of our contributions are mentioned below.

(1) By representing the row and column vectors, we propose a novel classifier, LGSLM, which captures the linear geometry structure of a label matrix. Further, we adjust the value of positive labels and linearly represent the relatively compact regions of the label matrix. To the best of our knowledge, we are the first to apply linear representation to multi-label learning.

(2) To reduce the computational complexity of row representation, after transforming the row representation optimization as a linear system, we decompose the system matrix into a block diagonal matrix and obtain the multiple

subsystems, which reduces the computational complexity from $O(l^6)$ to $O(l^4)$ (refer to **Theorem 1**).

(3) Experiments on five real-world datasets show that our approach has the best prediction performance, which means that the linear representation of a label matrix is more intrinsic than the low-rank constraint.

2 Related Work

As mentioned in the previous section, under the low-rank assumption, many existing approaches capture the label correlation using linear embedding methods. For example, based on principal component analysis, Tai et al. [15] proposed principal label space transformation by compressing the label vector. Further, Zhang et al. [21] simultaneously compressed the label space and feature space using canonical correlation analysis that extract the most predictable directions in the label space to characterize the label dependency. Subsequently, to make output codes predictable and discriminative, Zhang et al. [22] developed maximum margin output coding (MMOC) to learn output coding that is not only close to the correct codeword but is also far from any correct codeword. MMOC requires an expensive decoding procedure to recover the multiple labels of each testing instance, which becomes unacceptable when there are many labels. To avoid decoding completely, Liu et al. [11] proposed large margin metric learning with k-nearest neighbor constraints. Yu et al. [18] decomposed the regression matrix into two low-rank matrices, compressed the high-dimension feature space into low-dimension space and learned the classifier on the low-dimension feature space. However, these approaches have one common problem: because the label matrix becomes sparse, the low-rank assumption can be easily violated, thereby leading to degradation in the testing performance.

To address this problem, after grouping all instances into clusters, Bhatia et al. [2] embedded the label space of every cluster into a small space by preserving the pairwise distances between a label vector and the nearest label vectors. This breaks the traditional low-rank assumption and boosts classification accuracy. Based on the framework of [2], [5] exploited the Skip-Gram Negative Sampling (SGNS) approach to embed the label space of every cluster into low-dimension space. Although the method of [2, 5] captured the distance structure of every cluster, the small embedding neglected the overall label correlation. To address this problem, by simultaneously embedding the overall label space into a low-dimensional latent space, Shen et al. [12] exploited the manifold regularization to preserve the semantic similarity structure between the embeddings in the hamming space and the original label space. Additionally, after extracting the additional sparse component from the label matrix as outliers, Xu et al. [17] captured the low-rank structure of the residual matrix, thus validating the classical low-rank principle in multi-label learning. For the long-tail recommendation, [10] decomposed the overall items into a low-rank part and a sparse part. The low-rank part and the sparse part are respectively exploited to recommend the short-head items and the long-tail items. However, the small embedding dimen-

sion used in [2, 12] not only ignored the overall structure of the label matrix but also led to information loss.

3 Proposed Model

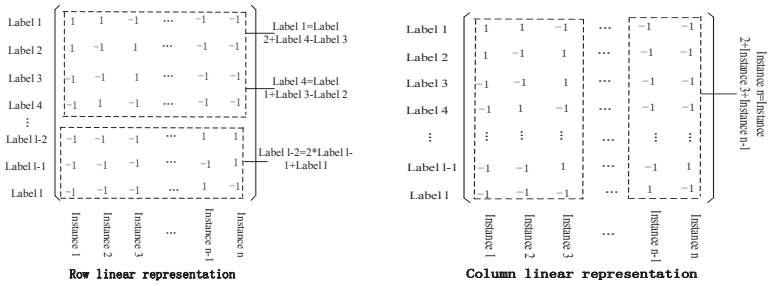

Fig. 1. In left figure, the row vector of label $l-2$ can be linearly represented by the row vectors of label $l-1$ and label l. In right figure, the column vector of instance n can be represented by the column vector of instance 2, instance 3 and instance $n-1$.

Let $\mathscr{D} = \{(\boldsymbol{x}_1, \boldsymbol{y}_1), (\boldsymbol{x}_2, \boldsymbol{y}_2), \cdots, (\boldsymbol{x}_n, \boldsymbol{y}_n)\}$ be the training data where instance $\boldsymbol{x}_i = (x_{i1}, \cdots, x_{id}) \in \mathbb{R}^d$ is a feature vector of d dimensions and $\boldsymbol{y}_i = (y_{i1}, \cdots, y_{il}) \in \{-1, +1\}^l$ is a label vector of l dimensions. $y_{ij} = 1$ means that instance $\boldsymbol{x}_i$ is associated with the j-th label (i.e. the positive label), $y_{ij} = -1$ means the j-th label does not exist in $\boldsymbol{x}_i$ (i.e. the negative label). In detail, given n training instances (represented by matrix $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n] \in \mathbb{R}^{d \times n}$) and label matrix (represented by matrix $\boldsymbol{Y} = [\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_n] \in \{-1, +1\}^{l \times n}$), our goal is to find an regression matrix $\boldsymbol{M} \in \mathbb{R}^{l \times d}$ to accurately predict the label vectors for unseen instances via computing the function $f(\boldsymbol{x}) = \boldsymbol{M}\boldsymbol{x}$. More specially, when $\boldsymbol{M}_i\boldsymbol{x}$ is greater than or equal to 0, the predicted i-th label of instance $\boldsymbol{x}$ is positive label; otherwise, the predicted i-th label of instance $\boldsymbol{x}$ is negative label.

3.1 Linear Representation of Row Vectors

The motivation for our framework comes from the following three observations. **Observation 1:** Every row/column vector of a label matrix can usually be linearly represented (or approximately represented) by the other row/column vectors. The linear representation, which does not depend on the rank structure of the label matrix, captures the linear correlation of several labels. **Observation 2:** The linear representation, which is usually obtained by minimizing certain distance of row/column vectors, has more influence to the negative labels because there exists plenty of negative labels and a small number of positive labels in dataset. **Observation 3:** After rearranging the columns of label matrix in decreasing order of the number of positive labels into a new matrix, the compact region of the new matrix has a more remarkable linear structure. In this section, we capture three observations by linearly representing the row/column vectors of the compact region of label matrix.

In linear representation, four critical factors (i.e. the number of row dimensions, the number of column dimensions, the number of positive labels and the

number of negative labels) have a strongly influence on the representative ability. For example, if the row dimension is not much less than the column dimension, one row vectors can usually be accurately represented by the other row vectors; when the number of positive labels is much less than the number of the negative labels, the linear representation have an more influence on the negative labels. Thus, by selecting the compact region which has the biggest fraction of positive labels to negative labels, we measure the relatively accurate linear representation discussed as follows.

After rearranging the columns of a label matrix in decreasing order according to the number of positive labels, without loss of generality, we assume that the new label matrix $\boldsymbol{Y}$ is segmented into the formulation $\boldsymbol{Y} = \{\boldsymbol{Y}_1^r, \boldsymbol{Y}_2^r, \cdots, \boldsymbol{Y}_S^r\}$ where the row is written as r, $\boldsymbol{Y}_s^r = \{\boldsymbol{y}_{s_1}, \boldsymbol{y}_{s_2}, \cdots, \boldsymbol{y}_{s_{k_s}}\}$, k_s is the number of instances from the s-th region, $\boldsymbol{Y}_i^r \cap \boldsymbol{Y}_j^r = \varnothing$ and $\bigcup_i \boldsymbol{Y}_i^r = \boldsymbol{Y}$.

For any local region $\boldsymbol{Y}_s^r$, the row linear representation $\boldsymbol{R}_s^r$ is as follows,

$$\boldsymbol{R}_s^r \boldsymbol{Y}_s^r = \boldsymbol{Y}_s^r, \boldsymbol{R}_{s_{(ii)}}^r = \boldsymbol{0}, ||\boldsymbol{R}_s^r||_2^2 < K_0. \tag{1}$$

where $||\boldsymbol{R}_s^r||_2^2$ is the regularization term and $\boldsymbol{R}_{s_{(ii)}}^r = \boldsymbol{0}$ denotes the diagonal elements to be 0, which eliminates the trivial solution of writing a label function as a linear combination of itself. In formulation $\boldsymbol{R}_s^r \boldsymbol{Y}_s^r = \boldsymbol{Y}_s^r$, $\boldsymbol{R}_s^r = [\boldsymbol{r_1}, \boldsymbol{r_2}, \cdots, \boldsymbol{r}_{l_s}] \in \mathbb{R}^{l_s \times l_s}$ is the linear representation of $\boldsymbol{Y}_s^r$, where l_s is the number of labels.

To guarantee the consistency between the row representation of real label matrix and the row representation of predicted label matrix, we constrain $\boldsymbol{R}_s^r \boldsymbol{M} \boldsymbol{X}_s^r = \boldsymbol{M} \boldsymbol{X}_s^r$ where $\boldsymbol{X}_s^r$ is the corresponding feature matrix to the local label matrix $\boldsymbol{Y}_s^r$. Here, we usually select the $\boldsymbol{Y}_1^r$ to capture the row linear representation.

If the row dimension is much less than the column dimension, row vectors may be not well represented by the other row vectors. Here, we capture the column linear representation to complementary the shortcoming. As the column dimension becomes larger, one column vector may be represented by small fraction of column vectors so that the column linear representation is sparse. Without loss of generality, we assume that the label matrix $\boldsymbol{Y}$ is segmented into the set $\boldsymbol{Y} = \{\boldsymbol{Y}_1^c, \boldsymbol{Y}_2^c, \cdots, \boldsymbol{Y}_P^c\}$ where the column is written as c, $\boldsymbol{Y}_p^c = \{\boldsymbol{y}_{p_1}, \boldsymbol{y}_{p_2}, \cdots, \boldsymbol{y}_{p_{k_p}}\}$, k_p is the number of the p-th set, $\boldsymbol{Y}_i^c \cap \boldsymbol{Y}_j^c = \varnothing$ and $\boldsymbol{Y}_i^c \cup \boldsymbol{Y}_j^c = \boldsymbol{Y}$. For any local region $\boldsymbol{Y}_p^c$, we have the following column linear representation $\boldsymbol{R}_p^c$,

$$\boldsymbol{Y}_p^c \boldsymbol{R}_p^c = \boldsymbol{Y}_p^c, \boldsymbol{R}_{p_{(ii)}}^c = \boldsymbol{0}, ||\boldsymbol{R}_k^c||_0 < K_1. \tag{2}$$

In formulation $\boldsymbol{Y}_p^c \boldsymbol{R}_p^c = \boldsymbol{Y}_p^c$, $\boldsymbol{Y}_p^c = [\boldsymbol{r_1}, \boldsymbol{r_2}, \cdots, \boldsymbol{r}_{l_p}] \in \mathbb{R}^{l \times l_p}$ is the linear correlation of $\boldsymbol{Y}_p^c$ where constraint $\boldsymbol{R}_{k_{ii}}^c = 0$ eliminates the trivial solution of writing a label function as a linear combination of itself. $||\boldsymbol{R}_p^c||_0$ is the number of non-zero elements of $\boldsymbol{R}_p^c$.

Since $||\boldsymbol{R}_p^c||_0$ tends to increase the optimization complexity, we employ one popular L1-norm minimization (to encourage sparsity) [4] to relax the constraints in problem (2), such that

$$\boldsymbol{Y}_p^c \boldsymbol{R}_p^c = \boldsymbol{Y}_p^c, \boldsymbol{R}_{p_{ii}}^c = \boldsymbol{0}, ||\boldsymbol{R}_p^c||_1 < K_1. \tag{3}$$

To guarantee the consistency between the linear representation of real label matrix and the linear representation of predicted label, we constrain the following equality,

$$Re = \{\boldsymbol{M}\boldsymbol{X}_p^c\boldsymbol{R}_p^c = \boldsymbol{M}\boldsymbol{X}_p^c | p \in P\}.$$

where $\boldsymbol{X}_p^c$ is the corresponding feature matrix to the local label matrix $\boldsymbol{Y}_p^c$.

3.2 The Strengthen of Linear Representation and Unified Objective Function

In the represented region, there exists plenty of negative labels and a small number of positive labels. In such situation, the linear representation, which is measured by minimizing a distance of row/column vectors (referred to problem (1)), has more influence on the negative labels than the positive labels. This usually makes the linear representation be only effective in relatively small region. To address the problem, we adjust the value of positive labels to effectively capture the linear geometry structure of larger region. Motivated by empirical observation, we model the weight of the positive label value as a sigmoidal function of $\log N_l$ [8],

$$\bar{\boldsymbol{Y}}_{ij} = 1 + Ce^{-Alog(N_l+B)}, if \quad \boldsymbol{Y}_{ij} = 1.$$
$$C = (logN - 1)(B + 1)^A$$

where A and B are two fixed parameters. In our experiment, we fix the value of A and B as 0.1.

We use the square loss to evaluate the discrepancy between $\bar{\boldsymbol{Y}}$ and the predicted label matrix $\boldsymbol{M}\boldsymbol{X}$. Combining the row linear representation and the column linear representation, converting the problem into an unconstrained optimization problem by using a penalty method, we obtain the following optimization problem,

$$\min_{\boldsymbol{M}} L(\bar{\boldsymbol{Y}}, \boldsymbol{M}\boldsymbol{X}) = ||\bar{\boldsymbol{Y}} - \boldsymbol{M}\boldsymbol{X}||_2^2 + \lambda_1||\boldsymbol{M}||_2^2$$
$$+\lambda_2 \sum_{s\in S} ||\boldsymbol{R}_s^r\boldsymbol{M}\boldsymbol{X}_s^r - \boldsymbol{M}\boldsymbol{X}_s^r||_2^2 + \lambda_3 \sum_{p\in P} ||\boldsymbol{M}\boldsymbol{X}_p^c\boldsymbol{R}_p^c - \boldsymbol{M}\boldsymbol{X}_p^c||_2^2 \qquad (4)$$

where λ_1 is an parameter to control regularization, penalty parameters $\lambda_2 > 0$, $\lambda_3 > 0$ are used to control the weight of row/column linear representation, respectively. Parameters λ_2 and λ_3 should be large to ensure that both $\boldsymbol{R}_s^r\boldsymbol{M}\boldsymbol{X}_s^r = \boldsymbol{M}\boldsymbol{X}_s^r$ and $\boldsymbol{M}\boldsymbol{X}_p^c\boldsymbol{R}_p^c = \boldsymbol{M}\boldsymbol{X}_p^c$ are satisfied.

4 Optimization Algorithm

In this section, we first present the optimization method for problem (1) and problem (3). Subsequently, we divide the objective function (4) into two cases and present the corresponding optimization methods.

4.1 Solving $\boldsymbol{R^r}$

Problem (1) can be transformed into the following form,

$$\min_{\boldsymbol{R}_s^r} ||\boldsymbol{R}_s^r \boldsymbol{Y}_s^r - \boldsymbol{Y}_s^r||_2^2 + c_1||\boldsymbol{R}_s^r||_2^2 + c_2 \sum_{i=1}^{l} ||\boldsymbol{R}_{s_{ii}}^r||_2^2. \tag{5}$$

where l is the number of labels and $(c_2 \to \infty)$ makes $\boldsymbol{R}_{s_{ii}}^r$ approach to 0, which avoids writing a linear representation as a linear combination of itself. Problem (5) can usually be transformed as solving one $l^2 \times l^2$ linear system, which requires the complexity of $O(l^6)$. To reduce the computation complexity, we decompose the linear system into l number of $l \times l$ linear sub-systems, which requires the complexity of $O(l^4)$.

Theorem 1. The problem (5) is equivalent to the set of sub-linear systems:

$$(2\boldsymbol{Y}_s^r \boldsymbol{Y}_s^{rT} + 2c_1\boldsymbol{I} + 2c_2\boldsymbol{I}_{ii})(\boldsymbol{R}_s^r)_{i,:}^T = (2\boldsymbol{Y}_s^r \boldsymbol{Y}_s^{rT})_{i,:}^T,$$
$$i = 1, 2, \cdots, l.$$

where $(-)_{i,:}$ is the i-th row of the corresponding matrix or vector, $(-)^T$ is the transposition of the corresponding matrix and vector, $\boldsymbol{I}_{ii}$ is an $l \times l$ matrix whose element are 0 except the ii entry is 1.

Thus, the solution of the linear system is $\mathbf{vec}(\boldsymbol{R}_s^r) = 2diag\{\boldsymbol{A}_1^{-1}, \boldsymbol{A}_2^{-1}, \cdot \cdot \cdot, \boldsymbol{A}_l^{-1}\}\mathbf{vec}(\boldsymbol{Y}_s^r \boldsymbol{Y}_s^{rT})$ where $\mathbf{vec}(\boldsymbol{R}_s^r)$ and $\mathbf{vec}(\boldsymbol{Y}_s^r \boldsymbol{Y}_s^{rT})$ are two vectors formed by the row vectors of $\boldsymbol{R}_s^r$ and $\boldsymbol{Y}_s^r \boldsymbol{Y}_s^{rT}$, $\boldsymbol{A}_i = 2\boldsymbol{Y}_s^r \boldsymbol{Y}_s^{rT} + 2c_1\boldsymbol{I} + 2c_2\boldsymbol{I}_{ii}$.

Proof: Taking the derivation of (5) w.r.t. $\boldsymbol{R}_s$ and setting it zero, we have the following form,

$$\boldsymbol{R}_s(2\boldsymbol{Y}_s\boldsymbol{Y}_s^T + 2c_1\boldsymbol{I}) + 2c_2 diag(R_{s_{i,i}}) = 2\boldsymbol{Y}_s\boldsymbol{Y}_s^T \tag{6}$$

Problem (6) is an $l^2 \times l^2$ linear system. Let $\boldsymbol{vec}(\boldsymbol{R}_s)$ be an vector formed by the row vectors of $\boldsymbol{R}_s$. In the problem, the i-th row of $\boldsymbol{R}_s$ forms the following equality $(\boldsymbol{R}_s)_{i,:}(2\boldsymbol{Y}_s\boldsymbol{Y}_s^T + 2c_1\boldsymbol{I}) + 2c_2 R_{ii} = (2\boldsymbol{Y}_s\boldsymbol{Y}_s^T)_{i,:}$. The equality is equivalent to the following sub-linear system,

$$(2\boldsymbol{Y}_s\boldsymbol{Y}_s^T + 2c_1\boldsymbol{I} + 2c_2\boldsymbol{I}_{ii})(\boldsymbol{R}_s)_{i,:}^T = (2\boldsymbol{Y}_s\boldsymbol{Y}_s^T)_{i,:}^T$$

The solution of the sub-linear system is $\boldsymbol{A}_i^{-1}(2\boldsymbol{Y}_s\boldsymbol{Y}_s^T)_{i,:}^T$ where $\boldsymbol{A}_i = (2\boldsymbol{Y}_s\boldsymbol{Y}_s^T + 2c_1\boldsymbol{I} + 2c_2\boldsymbol{I}_{ii})$. By traversing all rows of linear representation $\boldsymbol{R}_s$, we obtain the block matrix $A = diag\{\boldsymbol{A}_1^{-1}, \boldsymbol{A}_2^{-1}, \cdots, \boldsymbol{A}_l^{-1}\}$ and the solution $\boldsymbol{vec}(\boldsymbol{R}_s) = 2diag\{\boldsymbol{A}_1^{-1}, \boldsymbol{A}_2^{-1}, \cdots, \boldsymbol{A}_l^{-1}\}\boldsymbol{vec}(\boldsymbol{Y}_s\boldsymbol{Y}_s^T)$.

4.2 Solving $\boldsymbol{R^c}$

Problem (3) can be transformed into the following form,

$$\min_{\boldsymbol{R}_p^c} ||\boldsymbol{Y}_p^c \boldsymbol{R}_p^c - \boldsymbol{Y}_p^c||_2^2 + c_1||\boldsymbol{R}_p^c||_2^2 + c_2 \sum_{i=1}^{n_p} ||\boldsymbol{R}_{p_{ii}}^c||_2^2 + c_3||\boldsymbol{R}_p^c||_1. \tag{7}$$

where n_p is the number of represented column vectors and ($c_2 \to \infty$) makes $\boldsymbol{R}_{p_{ii}}$ approach to 0. In problem (7), the non-smoothness of $||\boldsymbol{R}_c^p||_1$ makes the standard proxy function-based optimization methods unsuitable. Thus, we exploit the alternating direction method of multipliers and soft thresholding operation [6] to optimize the problem.

By introducing an auxiliary variable $\boldsymbol{R}_c^p = \boldsymbol{Z}_1$, we transform problem (7) into the following formulation,

$$\begin{aligned}\min_{\boldsymbol{R}_p^c, \boldsymbol{Z}_1, \Upsilon_1} &||\boldsymbol{Y}_p^c \boldsymbol{R}_p^c - \boldsymbol{Y}_p^c||_2^2 + c_1||\boldsymbol{R}_p^c||_2^2 + c_2 \sum_{i=1}^{n_k} ||\boldsymbol{R}_{p_{ii}}^c||_2^2 \\ &+\beta||\boldsymbol{R}_p^c - \boldsymbol{Z}_1 + \Upsilon_1||_2^2 + c_3||\boldsymbol{Z}_1||_2 \end{aligned} \tag{8}$$

where $\Upsilon_1 \in \mathbb{R}^{l \times n_k}$ is the positive Lagrange multiplier, β is a positive number. Our method includes three key steps (i.e., updating $\boldsymbol{R}_p^c$, $\boldsymbol{Z}_1$, Υ_1, respectively).

Update $\boldsymbol{R}_p^c$ If $\boldsymbol{Z}_1$ and Υ_1 is fixed, $\boldsymbol{R}_p^c$ is updated by the following problem,

$$\min_{\boldsymbol{R}_p^c} ||\boldsymbol{Y}_p^c \boldsymbol{R}_p^c - \boldsymbol{Y}_p^c||_2^2 + c_1||\boldsymbol{R}_p^c||_2^2 + c_2 \sum_{i=1}^{n_k} ||\boldsymbol{R}_{p_{ii}}^c||_2^2 + \beta||\boldsymbol{R}_p^c - \boldsymbol{Z}_1 + \Upsilon_1||_2 \tag{9}$$

Taking the derivation of (9) w.r.t. $\boldsymbol{R}_p^c$ and setting it zero, we obtain the form $(2\boldsymbol{Y}_p^{cT}\boldsymbol{Y}_p^c + 2c_1\boldsymbol{I})\boldsymbol{R}_p^c + 2c_2 diag(\boldsymbol{R}_p^c) = 2\boldsymbol{Y}_p^{cT}\boldsymbol{Y}_p^c + 2\beta(\boldsymbol{Z}_1 - \Upsilon_1)$ which can be solved by the **Theorem 1** after transposing it.

Update $\boldsymbol{Z}_1$ If $\boldsymbol{R}_p^c$ and Υ_1 is fixed, $\boldsymbol{Z}_1$ is updated by solving the following problem,

$$\min_{\boldsymbol{Z}_1} \beta||\boldsymbol{R}_p^c - \boldsymbol{Z}_1 + \Upsilon||_1 + c_3||\boldsymbol{Z}_1||_1$$

which can be re-written into a convenient form:

$$\min_{\boldsymbol{Z}_1} ||\boldsymbol{R}_p^c - \boldsymbol{Z}_1 + \Upsilon||_1 + \frac{c_3}{\beta}||\boldsymbol{Z}_1||_1$$

The optimal $\boldsymbol{Z}_1$ can be obtained via soft thresholding operation $\boldsymbol{Z} = soft(\boldsymbol{R}_p^c + \Upsilon, \frac{c_3}{\beta})$ [6] where $soft(a, b) = sign(a)max(|a| - b, 0)$.

Updating $\boldsymbol{\Upsilon_1}$ If $\boldsymbol{R}_p^c$ and $\boldsymbol{Z}_1$ is fixed, Υ_1 is updated by the following form,

$$\Upsilon_1 = \Upsilon_1 + (\boldsymbol{R}_p^c - \boldsymbol{Z}_1)$$

4.3 The Optimization of Objective Function

In this section, we divide the objective into two cases. In case 1, we only consider the row representation of overall datasets. In case 2, we consider the mixture of row and column representation. In the following, we illustrate the optimization of two cases, respectively.

Row Representation of Overall Dataset When the linear representation only contains the row representation of overall datasets, the optimization problem is the following formulation,

$$\min_{\boldsymbol{M}} L(\bar{\boldsymbol{Y}}, \boldsymbol{MX}) = ||\bar{\boldsymbol{Y}} - \boldsymbol{MX}||_2^2 + \lambda_1||\boldsymbol{M}||_2^2 + \lambda_3||(\boldsymbol{R}^r - \boldsymbol{I})\boldsymbol{MX}||_2^2 \quad (10)$$

Taking the derivation of (10) w.r.t. $\boldsymbol{M}$ and setting it zero, we obtain the form $(2\boldsymbol{I} + 2\lambda_3(\boldsymbol{R}^r - \boldsymbol{I})^T(\boldsymbol{R}^r - \boldsymbol{I}))\boldsymbol{MXX}^T + 2\lambda_1\boldsymbol{M}$. Due to symmetry of $\boldsymbol{Q}_1 = 2\boldsymbol{I} + 2\lambda_2\boldsymbol{I} + 2\lambda_3(\boldsymbol{R}^r - \boldsymbol{I})^T(\boldsymbol{R}^r - \boldsymbol{I})$ and $\boldsymbol{XX}^T$, we decompose $\boldsymbol{Q}_1$ and $\boldsymbol{XX}^T$ into $\boldsymbol{Q}_1 = \boldsymbol{UAU}^T$, $\boldsymbol{XX}^T = \boldsymbol{VBV}^T$ where $\boldsymbol{U}$, $\boldsymbol{V}$ are the eigen vectors, and $\boldsymbol{A}$ and $\boldsymbol{B}$ are diagonal matrices. By substituting $\boldsymbol{Q}_1$ and $\boldsymbol{2XX}^T$, and multiplying $\boldsymbol{U}^T$ and $\boldsymbol{V}$ from left to right on both sides, we obtain the following form,

$$\boldsymbol{AU}^T\boldsymbol{MVB} + 2\lambda_1\boldsymbol{U}^T\boldsymbol{MV} = 2\boldsymbol{U}^T\bar{\boldsymbol{Y}}\boldsymbol{X}^T\boldsymbol{V} \quad (11)$$

Let $\boldsymbol{U}^T\boldsymbol{MV} = \overline{\boldsymbol{M}}$, then (11) becomes $\boldsymbol{A}\overline{\boldsymbol{M}}\boldsymbol{B} + 2\lambda_1\overline{\boldsymbol{M}} = \boldsymbol{U}^T\boldsymbol{QV}$. Thus, we can obtain $\boldsymbol{M}$ as $\boldsymbol{M} = \boldsymbol{U}\overline{\boldsymbol{M}}\boldsymbol{V}^T$ where $\overline{\boldsymbol{M}}_{ij} = \frac{(\boldsymbol{U}^T\boldsymbol{QV})_{ij}}{\boldsymbol{A}_{ii}\boldsymbol{B}_{jj}+2\lambda_1}$ denotes the (i,j)-th entry of $\overline{\boldsymbol{M}}$.

The Mixture of Row and Column Representation By introducing an auxiliary variable $\boldsymbol{M} = \boldsymbol{Z}_3$, we transform the objective function into the following formulation,

$$\begin{aligned}\min_{\boldsymbol{M}} L(\bar{\boldsymbol{Y}}, \boldsymbol{MX}) &= ||\bar{\boldsymbol{Y}} - \boldsymbol{MX}||_2^2 + \lambda_1||\boldsymbol{M}||_2^2 + \lambda_3||(\boldsymbol{R}_s^c - \boldsymbol{I})\boldsymbol{Z}_3\boldsymbol{X}_s||_2^2 \\ &+\lambda_3\sum_{p\in P}||\boldsymbol{MX}_p\boldsymbol{R}_p^c - \boldsymbol{MX}_p||_2^2 + \frac{\beta}{2}||\boldsymbol{M} - \boldsymbol{Z}_3||_2^2 + tr(\Upsilon_3(\boldsymbol{M} - \boldsymbol{Z}_3)) \quad (12)\end{aligned}$$

Solving M: By fixing the variables $\boldsymbol{Z}_3$ and Υ_3, taking the derivation of (12) w.r.t. $\boldsymbol{M}$, and setting the derivates zero, we have the form $\boldsymbol{M}((2+2\lambda_2)\boldsymbol{XX}^T + 2\lambda_3\boldsymbol{I} + 2\lambda_3\sum_{p\in P}\boldsymbol{X}(\boldsymbol{R}_p^c - \boldsymbol{I})(\boldsymbol{R}_p^c - \boldsymbol{I})^T\boldsymbol{X}^T) = 2\bar{\boldsymbol{Y}}\boldsymbol{X}^T$. And the solution is $\boldsymbol{M} = (2\bar{\boldsymbol{Y}}\boldsymbol{X}^T)((2+2\lambda_2)\boldsymbol{XX}^T + 2\lambda_3\boldsymbol{I} + 2\lambda_3\sum_{p\in P}\boldsymbol{X}_p\,(\boldsymbol{R}_p^c - \boldsymbol{I})(\boldsymbol{R}_p^c - \boldsymbol{I})^T\boldsymbol{X}^T)^{-1}$.

Solving Z_3**:** By fixing the variables $\boldsymbol{M}$ and Υ_3, we have the following formulation,

$$\min_{\boldsymbol{Z}_3} L(\bar{\boldsymbol{Y}}, \boldsymbol{MX}) = \lambda_3||(\boldsymbol{R}_s^r - \boldsymbol{I})\boldsymbol{Z}_3\boldsymbol{X}_i||_2^2 + \frac{\beta}{2}||\boldsymbol{M} - \boldsymbol{Z}_3||_2^2 + tr(\Upsilon_3(\boldsymbol{M} - \boldsymbol{Z}_3))$$

Take the derivation of (4.3) w.r.t. $\boldsymbol{Z}_3$. By setting the derivates zero, we have the similar formulation with Case 1.

Update Υ_3**:** By fixing the variables $\boldsymbol{Z}_3$ and $\boldsymbol{M}$, we can update Υ_3 by $\Upsilon_3 = \Upsilon_3 + \beta(\boldsymbol{M} - \boldsymbol{Z}_3)$.

5 Generalization Error Bounds

In this section we analyze excess risk bounds for our learning model with the row representation of overall dataset. Our model is characterized by a distribution $\mathscr{D}$ on the feature space and the label space $X \times Y$. Let n samples

$\{(\boldsymbol{x}_1, \boldsymbol{y}_1), (\boldsymbol{x}_2, \boldsymbol{y}_2), \cdots, (\boldsymbol{x}_n, \boldsymbol{y}_n)\}$ be drawn i.i.d. from the distribution $\mathscr{D}$, which is denoted by $D \sim D^n$.

Given this training data, we perform ERM to learn $\widehat{M} \in M$ by the following formulation,

$$\inf_{\boldsymbol{RMX}=\boldsymbol{MX}} \widehat{L}(M) = \frac{1}{n}\sum_{k=1}^{n} \mathscr{L}(y_i, f(x_i, \boldsymbol{M}))$$

where $\widehat{L}(M)$ is the empirical risk of the predictor $\boldsymbol{M}$. Our goal would be to show that $\widehat{M}$ has good generalization properties, i.e.

$$\inf L(\widehat{\boldsymbol{M}}) \leq \inf_{\boldsymbol{RMX}=\boldsymbol{MX}} \widehat{L}(\boldsymbol{M}) + \epsilon.$$

where $L(M) = \mathbb{E}_{x,y,l}[\![L(y_i^l, f_i^l(x_i; M))]\!]$ is the population risk of a predictor.

The Rademacher complexity, which is an effective way to measure the complexity of a function class, is exploited to find the generalization error bound of the learning algorithm by standard approaches [1].

Definition 1. Let $X = \{x_1, \cdots, x_n\} \in X^n$ be independent samples and F be a class of functions mapping from X to R. The empirical Rademacher compxlexity of F is defined as,

$$\widehat{R}_n(F) = \mathbb{E}_\sigma[\sup_{f\in F} |\frac{2}{n}\sum_{j=1}^{n} \sigma_i f(x_i)| x_1, \cdots, x_n]$$

where $\sigma = (\sigma_1, \cdots, \sigma_n)$ are independent uniform 1-valued random variables. The Rademacher compxlexity of F is,

$$\widehat{R}_n(F) = \mathbb{E}_{x,\sigma}[\sup_{f\in F} |\frac{2}{n}\sum_{j=1}^{n} \sigma_i f(x_i)]$$

Thus, the Rademacher compxlexity of our algorithm is as follows,

$$\widehat{R}_n(F) = \frac{2}{n}\mathbb{E}_{x,\sigma}[\sup_{\boldsymbol{M}\in F} \sum_{i=1}^{n} \sigma_i \sum_{j=1}^{l} \boldsymbol{M}_{j,:} x_i]$$

where $\boldsymbol{M}_{j,:}$ is the j-th row of $\boldsymbol{M}$. Let $\widehat{x}$ denote the weighted summarization $\sum_{i=1}^{L} \sigma_i x_i$ and $\widehat{X}$ be the stacker of L copies of $\widehat{x}$. Thus, we simply the the multi-label Rademacher complexity as,

$$\widehat{R}_n(F) = \frac{2}{n}\mathbb{E}_{x,\sigma}[\sup_{M\in F} M\widehat{X}]$$

Theorem 2. The proposed algorithm learns $\boldsymbol{M}$ over n training points with l labels i.i.d. sampled from distribution $\mathscr{D} = \boldsymbol{X} \times \boldsymbol{Y}$. For any data point, we assume that $||x|| < \Lambda$. The learning algorithm encourages $\boldsymbol{RMX} = \boldsymbol{MX}$ which

Datasets	#Dataset	#Training	#Testing	#Features	#Labels	#Card-Label	#Domain
CAL500	502	452	50	68	174	26.044	music
COREL5K	5,000	4,500	500	499	374	3.522	images
DELICIOUS	16,105	14,495	1,610	500	983	9.020	text
EUR-Lex	19,348	17,413	1,935	5000	3,993	5.310	text
NUS-WIDE-V	269,648	161,789	117,859	128	81	1.869	images

Table 1. Statistics of five datasets.

usually satisfies the $||\boldsymbol{RMX} - \boldsymbol{MX}||_2^2 < \epsilon_0$ (ϵ_0 is a constant). We also assume $(\boldsymbol{R}-\boldsymbol{I})^{-1}$ to be upper bounded by δ. Then, the Rademacher complexity of F is,

$$\widehat{R}_n(F) \leq \frac{2\delta\epsilon}{n}.$$

Proof. The learning algorithm constrains $\boldsymbol{RMX} = \boldsymbol{MX}$ to capture the linear representation, which is usually not satisfied. Instead of equation constraint, we have $||\boldsymbol{RMX}-\boldsymbol{MX}||_2^2 < \epsilon_0$ been satisfied. By multiplying $(\boldsymbol{R}-\boldsymbol{I})^{-1}$ left on $||\boldsymbol{RMX} - \boldsymbol{MX}||_2^2 < \epsilon_0$, we have the following formulation $||\boldsymbol{MX}||_2^2 < (\boldsymbol{R}-\boldsymbol{I})^{-1}\epsilon_0$. According to Eq. $\widehat{R}_n(F) = \frac{2}{n}\mathbb{E}_{x,\sigma}[\sup_{M\in F} M\widehat{X}]$, we prove $\widehat{R}_n(F) \leq \frac{2||(\boldsymbol{R}-\boldsymbol{I})^{-1}||\epsilon}{n}$.

6 Experiment

In this section, we perform experiments to evaluate the effectiveness of our method. After introducing datasets and methodology, we compare our method with two representative class of methods.

6.1 Datasets

The experiments are conducted on five real-world multi-label datasets[5], namely CAL500, COREL5K, DELICIOUS, EUR-Lex, NUS-WIDE-V.

1) Cal500: In this dataset, 4500 songs by different artists are annotated with 174 labels to represent their genres, instruments, emotions, and other related concepts [16].

2) COREL5K: It contains about 5000 images from Stock Photo CDs which are annotated with 374 labels [3].

3) DELICIOUS: It contains textual data of Web pages along with 983 tags extracted from the del.icio.us social book marking site [7].

4) EUR-Lex: It collects documents on European Union law [23]. There are several EuroVoc descriptors, directory codes, and types of subject matter to describe the labels.

5) NUS-WIDE-V: In this version, it represents the images as 128-D cVLAD+ features described in [14].

[5] http://mulan.sourceforge.net/datasets-mlc.html

Datasets	Example-F1 ↑							
	CCA	kNN	ML-kNN	LM-kNN	SLEEC	CoH	RROD(ours)	MRCR(ours)
CAL500	0.3520	0.3561	0.3216	0.3511	0.3099	0.3602	0.4501	**0.4606**
COREL5K	N/A	0.0223	0.0178	0.1295	0.0839	0.1996	0.2721	**0.2943**
DELICIOUS	N/A	0.1878	0.1518	0.2553	0.2197	0.2745	**0.4024**	0.3645
EUR-Lex	N/A	0.3409	0.3005	0.3818	0.3705	0.3915	**0.4257**	0.4252
NUS-WIDE-V	N/A	0.1382	0.0331	0.0982	0.1703	0.2838	0.3428	**0.3429**
Datasets	MIcro-F1↑							
	CCA	kNN	ML-kNN	LM-kNN	SLEEC	CoH	RROD(ours)	MRCR(ours)
CAL500	0.3537	0.3593	0.3184	0.3542	0.3077	0.3664	0.4500	**0.4607**
COREL5K	N/A	0.0321	0.0278	0.1670	0.1215	0.2127	0.2675	**0.2929**
DELICIOUS	N/A	0.2154	0.1738	0.3104	0.2532	0.3011	**0.4044**	0.3626
EUR-Lex	N/A	0.4011	0.3489	0.4344	0.4235	**0.4689**	0.0933	0.0348
NUS-WIDE-V	N/A	0.2772	0.2826	0.2093	0.3249	0.3282	0.3634	**0.3638**
Datasets	Macro-F1 ↑							
	CCA	kNN	ML-kNN	LM-kNN	SLEEC	CoH	RROD(ours)	MRCR(ours)
CAL500	0.0917	0.0971	0.0534	0.1001	0.0462	0.1173	**0.1826**	0.1698
COREL5K	N/A	0.052	0.0086	0.0266	0.0240	0.0423	**0.0483**	0.0433
DELICIOUS	N/A	0.0550	0.0481	0.1383	0.0702	0.0993	**0.1851**	0.1128
EUR-Lex	N/A	0.0877	0.0635	0.0919	0.0805	**0.1053**	0.0376	0.0212
NUS-WIDE-V	N/A	0.0682	0.0159	0.0171	0.0437	0.0498	**0.0809**	0.0803

Table 2. Predictive performance comparison on five real-world datasets with the embedding method of label matrix. The best results are in bold. N/A denotes not available.

6.2 Methodology

Baselines:

To demonstrate the performance of our method, we compare it with two representative approaches: the label-matrix embedding method and the regression-matrix constraining method. The label-matrix embedding method contains CCA (Zhang and Schneider 2011) [21], kNN, ML-kNN (Zhang and Zhou 2007) [19], LM-kNN (Liu and Tsang 2015) [11], SLEEC (Bhatia et al. 2015) [2], COH (Xiaobo Shen et al. 2018) [12]. The regression-matrix constraining method contains LEML(Yu et al. 2014) [18], SLRM (Jing et al. 2015) [9], REML(Xu et al. 2016) [17]. In LM-kNN, we set $\eta = 0.4$ and $C = 10$ by following the experimental settings [11]. In SLEEC, following the original settings [2], we set the number of clusters as $\frac{n}{6000}$ and the number of learners as 15. In COH [12], by following the original settings, the regularization parameter is empirically set to 100 for the two NUS-WIDE datasets, and 1 for the other datasets. In LEML [18], we set the rank as 50, 30, 170,1100, 70, 70 on datasets CAL, CORE5K, DELICIOUS, EUR-Lex, NUS-WIDE-V. In SLRM (Jing et al. 2015), we consider the N nearest neighbor graph of every instance to model the distance structure among data. In REML [2], we set the rank of regression matrix to be similar to LEML [18].

Evaluation Metric: Three metrics (i.e., Example-F1, Micro-F1, Macro-F1) are often used to evaluate the performance of multi-label learning and their detailed definitions can be found in Ref [13].

6.3 Experimental Results

Experimental settings. In this experiment, we consider two schemes: the row representation of overall datasets (RROD) and the mixture of row and column representation (MRCR). In RROD, we set the parameters $\lambda_1 = 1$ and $\lambda_2 = 1.3$ on every dataset. By adding the column representation into RROD, we obtain the MRCR and set the parameters $\lambda_1 = 1$, $\lambda_2 = 1.3$ and $\lambda_3 = 5$ on every datasets. More specially, on CAL dataset, the column representation of overall dataset is computed. In dataset CORE, DELICIOUS and EUR-Lex, the one-third of instances are selected to compute the column representation. In NUS-WIDE-V, we only select 1500 number of instances to compute the column representation.

Experimental Analysis on Embedding Methods In Table 1, in term of Example-F1, Micro-F1, Macro-F1, we present the prediction performances of CCA, kNN, ML-kNN, LM-kNN, SLEEC, COH on CAL500, COREL5K, DELICIOUS, EUR-Lex and NUS-WIDE-V. The best results of each approach are marked in bold.

From this table, we can see that, our approach generally has the best performance on the five datasets. For example, on the CAL500 dataset, in term of Example-F1, Micro-F1, Macro-F1, our approach improves the best results of the baselines by 10.04%, 9.43% and 6.53%. This means that the linear representation may capture the intrinsic geometry structure of label matrix. The detailed descriptions are as follows.

1) Our schemes outperform than the SLEEC and COH which both break the low-rank structure of label matrix by preserving the distance structure of label matrix. This shows that linear representation may keep the more information than the small embedding in SLEEC and COH.

2) In EUR-Lex and NUS-WIDE-V, the performance of MRCR does not outperform than RROD. This shows that the column representation may be covered by row representation, thus the column representation does not work.

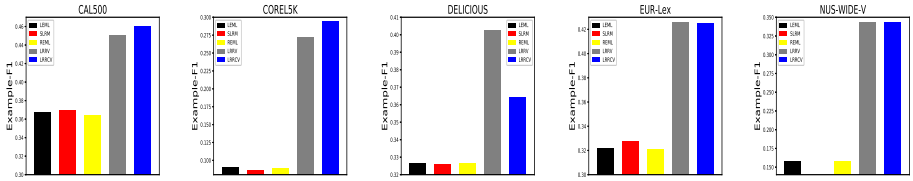

Fig. 2. Predictive performance of comparison in term of Example-F1 on five real-datasets with the method of low-rank constraint on regression matrix.

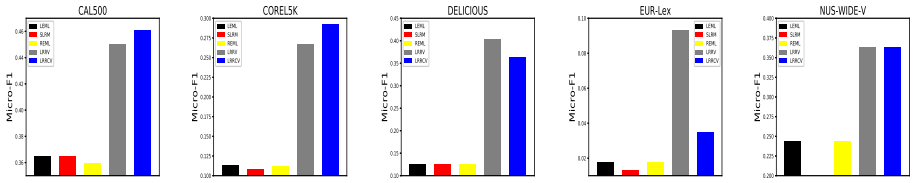

Fig. 3. Predictive performance of comparison in term of Micro-F1 on five real-datasets with the method of low-rank constraint on regression matrix.

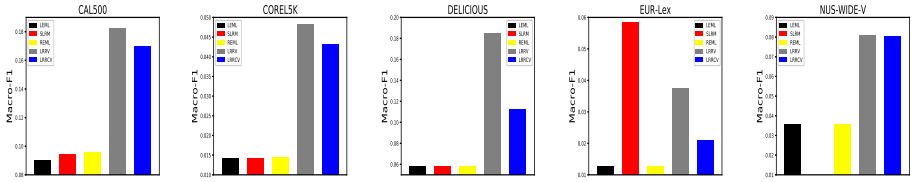

Fig. 4. Predictive performance of comparison in term of Macro-F1 on five real-datasets with the method of low-rank constraint on regression matrix.

Experimental Analysis on Low-rank constraint In this section, we compare the proposed RROD and MRCR methods with LEML[18], SLRM [9], REML [17], which directly capture the rank structure of regression matrix. In REML, we do not evaluate the performance of NUS-WIDE-V because a $N \times N$ matrix is too large to be stored. Figure 2, Figure 3 and Figure 4 respectively show the prediction performance of Example-F1, Micro-F1, and Macro-F1 on five datasets. From the three Figures, we can see that, our approach usually has the best performance on five datasets. This again shows that linear representation may keep more information than the embedding of feature space. This shows that our scheme outperforms than the methods of the rank-constraint of regression matrix.

7 Conclusion

We propose the LGSLM classifier to capture the intrinsic geometry structure of label matrix. More specially, we measure the row linear representation of overall datasets. In the column representation, by segmenting the label matrix into subregions and constraining the L_1 norm of the linear representation, we obtain the sparse structure of linear representation. Experiment results on five real datasets show the superior performances of our approach.

8 Acknowledgement

This work is supported by the National Key Research and Development Program of China (No. 2016QY06X1203), the Strategic Priority Research Program of the Chinese Academy of Sciences (No.XDC02040400), the National Natural Science Foundation of China (No.U1836203) and Shandong Provincial Key Research and Development Program (2019JZZY20127).

References

1. Bartlett, P.L., Mendelson, S.: Rademacher and Gaussian complexities: Risk bounds and structural results. Journal of Machine Learning Research. 3(Nov), pp.463-482 (2002).

2. Bhatia, K., Jain, H., Kar, P., Varma, M., Jain, P.: Sparse local embeddings for extreme multi-label classification. In: Proceedings of NeurIPS, pp.730-738 (2015).
3. Duygulu, P., Barnard, K., Freitas's, J., Forsyth, D.A.: Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In: Proceedings of ECCV, pp.97-112 (2002).
4. Elhamifar, E., Vidal, R.: Sparse subspace clustering. In: Proceedings of ICCV, pp.2790-2797 (2002).
5. Gupta, V., Wadbude, R., Natarajan, N., Karnick, H., Jain, P, Rai, P.: Distributional Semantics Meets Multi-Label Learning. In: Proceedings of AAAI, pp.3747–3754 (2019).
6. Donoho, D.L.: De-noising by soft-thresholding. IEEE transactions on information theory. 41(3), pp.613-627 (1995).
7. Tsoumakas, G., Katakis, I.: Multi-label classification: An overview. International Journal of Data Warehousing and Mining. 3(3), pp.1-13 (2007).
8. Jain, H., Prabhu, Y., Varma, M.: Extreme multi-label loss functions for recommendation, tagging, ranking and other missing label applications. In: Proceedings of SIGKDD, pp.935-944 (2016).
9. Jing, L., Yang, L.: Semi-supervised low-rank mapping learning for multi-label classification. In: Proceedings of CVPR, pp.1483-1491 (2015).
10. Li, J., Lu, K., Huang, Z., Shen, H.Y.: Two birds one stone: on both cold-start and long-tail recommendation. In: Proceedings of ACM MM, pp.898-906 (2017).
11. Liu, W., Tsang, I.W.: Large margin metric learning for multi-label prediction. In: Proceedings of AAAI, 2015.
12. Shen, X., Liu, W., Tsang, I.W., Sun, Q.S., Ong, Y.S.: Compact multi-label learning. In: Proceedings of AAAI, 2018.
13. Sorower, M.S.: A literature survey on algorithms for multi-label learning. Oregon State University. 18, pp.1-25 (2010).
14. Spyromitros, X.E., Papadopoulos, S., Kompatsiaris, I.Y., Tsoumakas, G., Vlahavas, I.: A comprehensive study over VLAD and product quantization in large-scale image retrieval. IEEE Transactions on Multimedia. 16(6), pp.1713-1728 (2014).
15. Tai, F., Lin, H.T.: Multi-label classification with principal label space transformation. Neural Computation. 24(9), pp.2508-2542 (2012).
16. Turnbull, D., Barrington, L., Torres, D., Lanckriet, G.: Semantic annotation and retrieval of music and sound effects. IEEE Transactions on Audio, Speech, and Language Processing. 16(2), pp.467-476 (2008).
17. Xu, C., Tao, D., Xu, C.: Robust extreme multi-label learning. In: Proceedings of SIGKDD, pp.1275-1284 (2016).
18. Yu, H.F., Jain, P., Kar, P., S. Dhillon, I.: Large-scale multi-label learning with missing labels. In: Proceedings of ICML, pp.593-601 (2014).
19. Zhang, M.L., Zhou, Z.H.: ML-KNN: A lazy learning approach to multi-label learning. Pattern recognition. 40(7), pp.2038-2048 (2007).
20. Zhang, M.L., Zhou, Z.H.: A review on multi-label learning algorithms. IEEE transactions on knowledge and data engineering. 26(8), pp.1819-1837 (2013).
21. Zhang, Y., Schneider, J.: Multi-label output codes using canonical correlation analysis. In: Proceedings of AISTATS, pp.873-882 (2011).
22. Zhang, Y., Schneider, J.: Maximum margin output coding. In: Proceedings of ICML, (2012).
23. Mencia, E.L., Frnkranz, J.: Efficient pairwise multilabel classification for large-scale problems in the legal domain. In Proceedings of Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pp.50-65 (2002).